Calculator.java:

```
package calculator;

import java.awt.*;

import javax.swing.*;

import java.awt.event.ActionListener;
import java.util.Stack;
import java.awt.event.ActionEvent;


public class Calculator {

	private JFrame frame;
	private KeyboardPanel keyBoardPanel;
	private HistoryPanel historyPanel;
	private EquationPanel equationPanel;
	private ScreenPanel screenPanel;
	boolean allowDot = true;
	double answer = 0;
	String state = "initial";

	/**
	 * Launch the application.
	 */
	public static void main(String[] args) {
		EventQueue.invokeLater(new Runnable() {
			public void run() {
				try {
					Calculator window = new Calculator();
					window.frame.setVisible(true);
				} catch (Exception e) {
					e.printStackTrace();
				}
			}
		});
	}
```

```java
/**
 * Create the application.
 */
public Calculator() {
	initialize();
}

/**
 * Initialize the contents of the frame.
 */
private void initialize() {
	frame = new JFrame();
	frame.setBounds(100, 100, 350, 700);
	frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
	frame.getContentPane().setLayout(null);

	keyBoardPanel = new KeyboardPanel();
	keyBoardPanel.setBounds(6, 445, 338, 267);
	frame.getContentPane().add(keyBoardPanel);
	keyBoardPanel.setLayout(null);

	historyPanel = new HistoryPanel();
	historyPanel.setBounds(6, 222, 338, 159);
	frame.getContentPane().add(historyPanel);
	historyPanel.setLayout(null);

	equationPanel = new EquationPanel();
	equationPanel.setBounds(6, 6, 338, 189);
	frame.getContentPane().add(equationPanel);
	equationPanel.setLayout(null);


	screenPanel = new ScreenPanel();
	screenPanel.setBounds(6, 393, 338, 47);
	frame.getContentPane().add(screenPanel);
	screenPanel.setLayout(null);

	initEquationListeners();
	initHistoryListeners();
	initKeyboardListeners();

}

private void initEquationListeners() {
    equationPanel.getAddButton().addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            // If this is a click
            String equation = "y = f(x) = " + equationPanel.getEquation();
            historyPanel.addToHistory(equation);
        }
    });

    equationPanel.getDeleteButton().addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            historyPanel.deleteFromHistory();
        }
    });

}

private void initHistoryListeners() {
    historyPanel.getLoadButton().addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            String equation = historyPanel.getEquationOnLoadButtonClick();
            equation = equation.substring(11, equation.length());
            equationPanel.setEquation(equation);
        }
    });
}

private void initKeyboardListeners() {

    keyBoardPanel.getClearButton().addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            screenPanel.setText("0");
            allowDot = true;
            state = "initial";
        }
    });

    keyBoardPanel.getDotButton().addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            if(allowDot) {
            String text = screenPanel.getText() + ".";
            screenPanel.setText(text);
            allowDot = false;
            }
        }
        });

        for(JButton numberButton: keyBoardPanel.getNumberButtons()) {
            numberButton.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e) {

                    String text = screenPanel.getText();
                    if(state.equals("initial")) {
                        if(String.valueOf(text.charAt(text.length()-
1)).equals("0")) {
                            text = "";

                        }
                    }
                    else if(state.equals("result") || state.equals("pi/e")) {
                        String lastChar =
String.valueOf(text.charAt(text.length()-1));
                        while(text.length() != 0 && (!lastChar.equals("+") ^
!lastChar.equals("-") ^ !lastChar.equals("*") ^ !lastChar.equals("/") ^ !lastChar.equals("^") )) {
                            text = text.substring(0, text.length()-1);
                            if (text.length() > 0) {
                                lastChar =
String.valueOf(text.charAt(text.length()-1));
                            }
                        }
                    }
                    text = text + numberButton.getText();
                    screenPanel.setText(text);
                    state = "number";
                }

            });
        }

        for (JButton valueButton: keyBoardPanel.getValueButtons()) {
            valueButton.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    String text = screenPanel.getText();
                    if(state.equals("initial")) {
                        if(String.valueOf(text.charAt(text.length()-
1)).equals("0")) {
                            text = "";

                        }
                    }
                    else if(state.equals("result") || state.equals("pi/e") ||
state.equals("number")) {
                        String lastChar =
String.valueOf(text.charAt(text.length()-1));
                        while(text.length() != 0 && (!lastChar.equals("+") ^
!lastChar.equals("-") ^ !lastChar.equals("*") ^ !lastChar.equals("/") ^ !lastChar.equals("^") )) {
                            text = text.substring(0, text.length()-1);
                            if (text.length() > 0) {
                                lastChar =
String.valueOf(text.charAt(text.length()-1));
                            }
                        }
                    }
                    String btnText = valueButton.getText();
                    if(btnText.equals("π")) {
                        btnText = "3.14";
                    }
                    else if(btnText.equals("e")) {
                        btnText = "2.72";
                    }
                    else if(btnText.equals("ANSWER")) {
                        btnText = String.valueOf(answer);
                    }
                    screenPanel.setText(text + btnText);
                    allowDot = false;
                    state = "pi/e";
                }
            });
        }

        for(JButton resultOperationButton: keyBoardPanel.getresultOperations()) {
            resultOperationButton.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    String text = screenPanel.getText();
                    double result =
Double.parseDouble(evaluateExpression(text));
                    String operation = resultOperationButton.getText();
                    switch(operation) {
                    case "sin":
                        result = Math.sin(Math.toRadians(result));
                        break;
                    case "cos":
                        result = Math.cos(Math.toRadians(result));
                        break;
                    case "tan":
                        result = Math.tan(Math.toRadians(result));
                        break;
                    case "ln":
                        result = Math.log(result);
                        break;
                    case "√":
                        result = Math.sqrt(result);
                        break;
                    }
                    answer = result;
                    String answer = String.valueOf(result);

                    screenPanel.setText(answer);
                    allowDot = false;
                    state = "result";
                }
            });
        }

        for(JButton operationButton: keyBoardPanel.getOperationButtons()) {
            operationButton.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    String text=screenPanel.getText();
                    String operation = operationButton.getText();
                    if(text.length()>0) {
                        String lastChar = 
String.valueOf(text.charAt(text.length()-1));
                        if(lastChar.equals("+") || lastChar.equals("-") || 
lastChar.equals("*") || lastChar.equals("/") || lastChar.equals("^") ) {
                            text = text.substring(0, text.length() - 1);
                            }
                    }
                    text = text + operation;
                    screenPanel.setText(text);
                    allowDot = true;
                    state = "operation";
                }
            });
        }
    }

    public String evaluateExpression(String equation) {
```

```
              Stack<Integer> op  = new Stack<Integer>();
Stack<Double> val = new Stack<Double>();
Stack<Integer> optmp  = new Stack<Integer>();
Stack<Double> valtmp = new Stack<Double>();
String input = "0" + equation;
input = input.replaceAll("-","+-");
/* Store operands and operators in respective stacks */
String temp = "";
for (int i = 0;i < input.length();i++)
{
   char ch = input.charAt(i);
   if (ch == '-')
      temp = "-" + temp;
   else if (ch != '+' &&  ch != '*' && ch != '/' && ch !='^')
     temp = temp + ch;
   else
   {
      val.push(Double.parseDouble(temp));
      op.push((int)ch);
      temp = "";
   }
}
val.push(Double.parseDouble(temp));
/* Create char array of operators as per precedence */
/* -ve sign is already taken care of while storing */
char operators[] = {'^', '/','*','+'};
/* Evaluation of expression */
for (int i = 0; i < 4; i++)
{
   boolean it = false;
   while (!op.isEmpty())
   {
      int optr = op.pop();
      double v1 = val.pop();
      double v2 = val.pop();
      if (optr == operators[i])
      {
         /* if operator matches evaluate and store in temporary stack */
              if( i == 0) {
                        valtmp.push(Math.pow(v2, v1));
                        it = true;
                        break;
              }
              else  if (i == 1)
```

```
          {
            valtmp.push(v2 / v1);
            it = true;
            break;
          }
          else if (i == 2)
          {
            valtmp.push(v2 * v1);
            it = true;
            break;
          }
          else if (i == 3)
          {
            valtmp.push(v2 + v1);
            it = true;
            break;
          }
        }
        else
        {
          valtmp.push(v1);
          val.push(v2);
          optmp.push(optr);
        }
      }
      /* Push back all elements from temporary stacks to main stacks */
      while (!valtmp.isEmpty())
        val.push(valtmp.pop());
      while (!optmp.isEmpty())
        op.push(optmp.pop());
      /* Iterate again for same operator */
      if (it)
        i--;
    }
    String result = val.pop().toString();
    return result;

       }
}
```

EquationPanel.java

```
package calculator;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;



public class EquationPanel extends JPanel {

	private JLabel lblEquation;
	private JTextField equationTextField;
	private JLabel lblYFx;
	private JButton btnAdd;
	private JButton btnDel;
	private JLabel lblSelectAColor;
	private JComboBox<String> colorComboBox;


	public EquationPanel() {
		addKeys();
	}

	public void addKeys() {
		lblEquation = new JLabel("Equation :-");
		lblEquation.setBounds(86, 18, 80, 22);
		this.add(lblEquation);

		equationTextField = new JTextField();
		equationTextField.setToolTipText("Enter f(x) here");
		equationTextField.setBounds(80, 41, 199, 26);
		this.add(equationTextField);
		equationTextField.setColumns(10);

		lblYFx = new JLabel("y = f(x) =");
}

        lblYFx.setBounds(16, 46, 61, 16);
        this.add(lblYFx);

        btnAdd = new JButton("ADD");
        btnAdd.setBounds(80, 73, 96, 35);
        this.add(btnAdd);

        btnDel = new JButton("DEL");
        btnDel.setBounds(183, 73, 96, 35);
        this.add(btnDel);

        JLabel lblSelectAColor = new JLabel("Select a color :-");
        lblSelectAColor.setBounds(80, 120, 123, 16);
        this.add(lblSelectAColor);

        String[] colors = {"BLACK", "BLUE", "GREEN", "RED", "YELLOW"};
        colorComboBox = new JComboBox<String>(colors);
        colorComboBox.setBounds(76, 139, 127, 27);
        this.add(colorComboBox);
        colorComboBox.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                        String color = colorComboBox.getSelectedItem().toString();
                        changeColorOfText(color);
                }
        });
}

public String getEquation() {
        return equationTextField.getText();
}

public void setEquation(String equation) {
        equationTextField.setText(equation);
}

public JButton getAddButton() {
        return btnAdd;
}

public JButton getDeleteButton() {
        return btnDel;
}

public void changeColorOfText(String color) {
        switch(color) {
        case "BLACK":
            equationTextField.setForeground(Color.BLACK);
            break;
        case "BLUE":
            equationTextField.setForeground(Color.BLUE);
            break;
        case "RED":
            equationTextField.setForeground(Color.RED);
            break;
        case "GREEN":
            equationTextField.setForeground(Color.GREEN);
            break;
        case "YELLOW":
            equationTextField.setForeground(Color.YELLOW);
            break;
            default:
                equationTextField.setForeground(Color.BLACK);
                break;
        }

    }

}
```

HistoryPanel.java

```
package calculator;

import javax.swing.*;
import java.awt.*;

public class HistoryPanel extends JPanel{
	private JButton btnLoad;
	private JList historyList;
	private JLabel lblHistory;
	private DefaultListModel<String> listModel;


	public HistoryPanel() {
		addKeys();
	}

	public void addKeys() {
		btnLoad = new JButton("LOAD");
		btnLoad.setBounds(63, 117, 93, 36);
		this.add(btnLoad);

		listModel =new DefaultListModel<String>();


		historyList = new JList(listModel);
		historyList.setBounds(66, 27, 204, 89);
		this.add(historyList);
		historyList.setSelectionMode(ListSelectionModel.SINGLE_INTERVAL_SELECTION);
		historyList.setLayoutOrientation(JList.VERTICAL);
		historyList.setVisibleRowCount(2);


		/*JScrollBar scrollBar = new JScrollBar();
		scrollBar.setBounds(255, 27, 15, 89);
		this.add(scrollBar);

		JScrollPane scrollPane = new JScrollPane();
		scrollPane.setViewportView(historyList);
		scrollPane.setBounds(66, 27, 204, 89);
		this.add(scrollPane);*/
		lblHistory = new JLabel("History :-");
		lblHistory.setBounds(66, 6, 75, 16);
		this.add(lblHistory);
	}

	public void addToHistory(String equation) {
		listModel.addElement(equation);

	}

	public void deleteFromHistory() {
		int index = historyList.getSelectedIndex();
		listModel.remove(index);
	}

	public JButton getLoadButton() {
		return btnLoad;
	}

	public String getEquationOnLoadButtonClick() {
		int index=historyList.getSelectedIndex();
		String equation = listModel.get(index);
		return equation;
	}
	/*public JList getHistoryList() {
		return historyList;
	}*/



}
```

KeyboardPanel.java

```
package calculator;

import javax.swing.*;
import java.awt.event.*;
import java.util.ArrayList;
import java.util.List;
;

public class KeyboardPanel extends JPanel {
	private JButton btnAnswer;
	private JButton btnEnter;

	private JButton btnDot;
	private JButton btn0;
	private JButton btnC;
	private JButton btnStar;
	private JButton btnPi;
	private JButton btnLn;

	private JButton btn3;
	private JButton btn2;
	private JButton btn1;
	private JButton btnDivide;
	private JButton btnE;
	private JButton btnTan;

	private JButton btn6;
	private JButton btn5;
	private JButton btn4;
	private JButton btnMinus;
	private JButton btnSqRt;
	private JButton btnCos;

	private JButton btn9;
	private JButton btn8;
	private JButton btn7;
	private JButton btnPlus;
	private JButton btnPower;
	private JButton btnSin;
```

```
ArrayList<JButton> numberButtonList;
ArrayList<JButton> operationButtonList;
ArrayList<JButton> resultOperations;
ArrayList<JButton> valueButtons;

public KeyboardPanel() {
        addAnswerKeys();
        addKeys();
}

public void addAnswerKeys() {
        btnAnswer = new JButton("ANSWER");
        btnAnswer.setBounds(44, 160, 117, 40);
        this.add(btnAnswer); //handle case
        valueButtons = new ArrayList<JButton>(3);
        valueButtons.add(btnAnswer);

        btnEnter = new JButton("ENTER");
        btnEnter.setBounds(160, 160, 117, 40);
        this.add(btnEnter);
        resultOperations = new ArrayList<JButton>(10);
        resultOperations.add(btnEnter);
}

public void addKeys() {

        // Add all keys here
        btnDot = new JButton(".");
        btnDot.setBounds(42, 120, 40, 40);
        this.add(btnDot); //Handle case

        btn0 = new JButton("0");
        btn0.setBounds(81, 120, 40, 40);
        this.add(btn0);
        numberButtonList = new ArrayList<JButton>(10);
        numberButtonList.add(btn0);

        btnC = new JButton("C");
        btnC.setBounds(120, 120, 40, 40);
        this.add(btnC);

        btnStar = new JButton("*");
        btnStar.setBounds(159, 120, 40, 40);
        this.add(btnStar);
```

```
operationButtonList = new ArrayList<JButton>(10);
operationButtonList.add(btnStar);

btnPi = new JButton("π");
btnPi.setBounds(198, 120, 40, 40);
this.add(btnPi);
//numberButtonList.add(btnPi); //Special case
valueButtons.add(btnPi);

btnLn = new JButton("ln");
btnLn.setBounds(237, 120, 40, 40);
this.add(btnLn);
resultOperations.add(btnLn);

//Row 2

btn3 = new JButton("3");
btn3.setBounds(42, 80, 40, 40);
this.add(btn3);
numberButtonList.add(btn3);

btn2 = new JButton("2");
btn2.setBounds(81, 80, 40, 40);
this.add(btn2);
numberButtonList.add(btn2);

btn1 = new JButton("1");
btn1.setBounds(120, 80, 40, 40);
this.add(btn1);
numberButtonList.add(btn1);

btnDivide = new JButton("/");
btnDivide.setBounds(159, 80, 40, 40);
this.add(btnDivide);
operationButtonList.add(btnDivide);

btnE = new JButton("e");
btnE.setBounds(198, 80, 40, 40);
this.add(btnE);
valueButtons.add(btnE);
//numberButtonList.add(btnE); //Special case

btnTan = new JButton("tan");
btnTan.setBounds(237, 80, 40, 40);
```

```
this.add(btnTan);
resultOperations.add(btnTan);

//Row 3

btn6 = new JButton("6");
btn6.setBounds(42, 40, 40, 40);
this.add(btn6);
numberButtonList.add(btn6);

btn5 = new JButton("5");
btn5.setBounds(81, 40, 40, 40);
this.add(btn5);
numberButtonList.add(btn5);

btn4 = new JButton("4");
btn4.setBounds(120, 40, 40, 40);
this.add(btn4);
numberButtonList.add(btn4);

btnMinus = new JButton("-");
btnMinus.setBounds(159, 40, 40, 40);
this.add(btnMinus);
operationButtonList.add(btnMinus);

btnSqRt = new JButton("√");
btnSqRt.setBounds(198, 40, 40, 40);
this.add(btnSqRt);
resultOperations.add(btnSqRt);

btnCos = new JButton("cos");
btnCos.setBounds(237, 40, 40, 40);
this.add(btnCos);
resultOperations.add(btnCos);

//Row 4

btn9 = new JButton("9");
btn9.setBounds(42, 0, 40, 40);
this.add(btn9);
numberButtonList.add(btn9);

btn8 = new JButton("8");
btn8.setBounds(81, 0, 40, 40);
```

```
}

        this.add(btn8);
        numberButtonList.add(btn8);

        btn7 = new JButton("7");
        btn7.setBounds(120, 0, 40, 40);
        this.add(btn7);
        numberButtonList.add(btn7);

        btnPlus = new JButton("+");
        btnPlus.setBounds(159, 0, 40, 40);
        this.add(btnPlus);
        operationButtonList.add(btnPlus);

        btnPower = new JButton("^");
        btnPower.setBounds(198, 0, 40, 40);
        this.add(btnPower);
        operationButtonList.add(btnPower);

        btnSin = new JButton("sin");
        btnSin.setBounds(237, 0, 40, 40);
        this.add(btnSin);
        resultOperations.add(btnSin);
}

public ArrayList<JButton> getNumberButtons() {
        return numberButtonList;
}

public ArrayList<JButton> getOperationButtons(){
        return operationButtonList;
}

public ArrayList<JButton> getresultOperations(){
        return resultOperations;
}

public JButton getClearButton() {
        return btnC;
}

public JButton getDotButton() {
        return btnDot;
}
```

```
        public ArrayList<JButton> getValueButtons(){
                return valueButtons;
        }



}
```

ScreenPanel.java

```
package calculator;

import javax.swing.*;
import java.awt.event.*;
import java.util.ArrayList;
import java.util.List;
;

public class KeyboardPanel extends JPanel {
	private JButton btnAnswer;
	private JButton btnEnter;

	private JButton btnDot;
	private JButton btn0;
	private JButton btnC;
	private JButton btnStar;
	private JButton btnPi;
	private JButton btnLn;

	private JButton btn3;
	private JButton btn2;
	private JButton btn1;
	private JButton btnDivide;
	private JButton btnE;
	private JButton btnTan;

	private JButton btn6;
	private JButton btn5;
	private JButton btn4;
	private JButton btnMinus;
	private JButton btnSqRt;
	private JButton btnCos;

	private JButton btn9;
	private JButton btn8;
	private JButton btn7;
	private JButton btnPlus;
	private JButton btnPower;
	private JButton btnSin;
```

```
ArrayList<JButton> numberButtonList;
ArrayList<JButton> operationButtonList;
ArrayList<JButton> resultOperations;
ArrayList<JButton> valueButtons;

public KeyboardPanel() {
	addAnswerKeys();
	addKeys();
}

public void addAnswerKeys() {
	btnAnswer = new JButton("ANSWER");
	btnAnswer.setBounds(44, 160, 117, 40);
	this.add(btnAnswer); //handle case
	valueButtons = new ArrayList<JButton>(3);
	valueButtons.add(btnAnswer);

	btnEnter = new JButton("ENTER");
	btnEnter.setBounds(160, 160, 117, 40);
	this.add(btnEnter);
	resultOperations = new ArrayList<JButton>(10);
	resultOperations.add(btnEnter);
}

public void addKeys() {

	// Add all keys here
	btnDot = new JButton(".");
	btnDot.setBounds(42, 120, 40, 40);
	this.add(btnDot); //Handle case

	btn0 = new JButton("0");
	btn0.setBounds(81, 120, 40, 40);
	this.add(btn0);
	numberButtonList = new ArrayList<JButton>(10);
	numberButtonList.add(btn0);

	btnC = new JButton("C");
	btnC.setBounds(120, 120, 40, 40);
	this.add(btnC);

	btnStar = new JButton("*");
	btnStar.setBounds(159, 120, 40, 40);
	this.add(btnStar);
```

```
operationButtonList = new ArrayList<JButton>(10);
operationButtonList.add(btnStar);

btnPi = new JButton("π");
btnPi.setBounds(198, 120, 40, 40);
this.add(btnPi);
//numberButtonList.add(btnPi); //Special case
valueButtons.add(btnPi);

btnLn = new JButton("ln");
btnLn.setBounds(237, 120, 40, 40);
this.add(btnLn);
resultOperations.add(btnLn);

//Row 2

btn3 = new JButton("3");
btn3.setBounds(42, 80, 40, 40);
this.add(btn3);
numberButtonList.add(btn3);

btn2 = new JButton("2");
btn2.setBounds(81, 80, 40, 40);
this.add(btn2);
numberButtonList.add(btn2);

btn1 = new JButton("1");
btn1.setBounds(120, 80, 40, 40);
this.add(btn1);
numberButtonList.add(btn1);

btnDivide = new JButton("/");
btnDivide.setBounds(159, 80, 40, 40);
this.add(btnDivide);
operationButtonList.add(btnDivide);

btnE = new JButton("e");
btnE.setBounds(198, 80, 40, 40);
this.add(btnE);
valueButtons.add(btnE);
//numberButtonList.add(btnE); //Special case

btnTan = new JButton("tan");
btnTan.setBounds(237, 80, 40, 40);
```

```
this.add(btnTan);
resultOperations.add(btnTan);

//Row 3

btn6 = new JButton("6");
btn6.setBounds(42, 40, 40, 40);
this.add(btn6);
numberButtonList.add(btn6);

btn5 = new JButton("5");
btn5.setBounds(81, 40, 40, 40);
this.add(btn5);
numberButtonList.add(btn5);

btn4 = new JButton("4");
btn4.setBounds(120, 40, 40, 40);
this.add(btn4);
numberButtonList.add(btn4);

btnMinus = new JButton("-");
btnMinus.setBounds(159, 40, 40, 40);
this.add(btnMinus);
operationButtonList.add(btnMinus);

btnSqRt = new JButton("√");
btnSqRt.setBounds(198, 40, 40, 40);
this.add(btnSqRt);
resultOperations.add(btnSqRt);

btnCos = new JButton("cos");
btnCos.setBounds(237, 40, 40, 40);
this.add(btnCos);
resultOperations.add(btnCos);

//Row 4

btn9 = new JButton("9");
btn9.setBounds(42, 0, 40, 40);
this.add(btn9);
numberButtonList.add(btn9);

btn8 = new JButton("8");
btn8.setBounds(81, 0, 40, 40);
```

```
}

    this.add(btn8);
    numberButtonList.add(btn8);

    btn7 = new JButton("7");
    btn7.setBounds(120, 0, 40, 40);
    this.add(btn7);
    numberButtonList.add(btn7);

    btnPlus = new JButton("+");
    btnPlus.setBounds(159, 0, 40, 40);
    this.add(btnPlus);
    operationButtonList.add(btnPlus);

    btnPower = new JButton("^");
    btnPower.setBounds(198, 0, 40, 40);
    this.add(btnPower);
    operationButtonList.add(btnPower);

    btnSin = new JButton("sin");
    btnSin.setBounds(237, 0, 40, 40);
    this.add(btnSin);
    resultOperations.add(btnSin);
}

public ArrayList<JButton> getNumberButtons() {
    return numberButtonList;
}

public ArrayList<JButton> getOperationButtons(){
    return operationButtonList;
}

public ArrayList<JButton> getresultOperations(){
    return resultOperations;
}

public JButton getClearButton() {
    return btnC;
}

public JButton getDotButton() {
    return btnDot;
}
```

```
	public ArrayList<JButton> getValueButtons(){
		return valueButtons;
	}



}
```